

## Feeding the Goose to Get Golden Eggs: Institutional Support and Legal Obstacles to Open Source Software

Bradford Castalia  
*Systems Analyst*  
Planetary Image Research Laboratory  
HiRISE Operations Center  
University of Arizona  
Tucson, Arizona



# Open Source Software

## Economy

- “Free” as in liberated (i.e. “open”)
  - The goose unconditionally lays eggs.*
- Guaranteed access
  - To the implementing source code
  - To modifications of the source code
- Use with closed source modules
  - Gnu Public License (GPL)
  - Limited (Library) GPL
  - Many licensing variations (<http://opensource.org/>)
  - Open source criteria for an open source license
- “Free” as in no price charged
  - The goose does not demand payment for the eggs.*
  - The source code has no price (it is “price-less”)
  - There may be vendor charges for media and other distribution costs



# Open Source Software

## Economy

- No price does not mean no cost

*The goose must be fed.*

- Development resources
  - Personnel, facilities, etc.
- Support
  - Distribution, maintenance, etc.
- Business model
  - Being a farmer raising geese.*
  - The value of the software is in its use
  - Sharing the costs
    - Giving back (fair dealing)
      - Who pays and who benefits?
    - Progressing forward (positive growth)
  - Services
    - Application and adaptation
    - Training



# Open Source Software

## Quality

- The value of free software

*The eggs are golden.*

- Recognizing the open source benefits
  - Measurable productivity and functionality benefits
  - Indirect benefits to the community and good will
- Cumulative benefits
  - Multiple perspectives
  - Refinements over time
- Open source vs. proprietary software: a false dichotomy

*Some eggs are rotten regardless of the kind of goose.*

  - Open source does not define production process
  - Evaluation for use remains a requirement
  - Guaranteed access ensures:
    - Transparency of code quality
    - Ability to implement fixes and enhancements
  - Developer support and/or user support



# Open Source Software

## Quality

- Security
  - Security by obscurity
    - Secrecy
    - The software is innocent until proven guilty
  - Reliability by rigor
    - Inspection is better than proscription
    - Many eyes see more defects
    - The FreeBSD example



# Open Source Software

## Policy

- Programs
  - Budget allocations for open source software support
  - Revise legal documents to facilitate use of open source software
    - Remove obstacles to access
    - Enable redistribution
    - Provide protection from financial liability
    - Focus on meeting user needs
- Procurement
  - Formal inclusion of open source software evaluations
  - Give preference to open source when all other criteria are met
- Production
  - Require publicly funded software to be open source
  - Establish a coordinated software capabilities library with a pro-active access and distribution system



# Open Source Software

## Policy

- Train software development staff in reusability techniques
  - Differentiate application specific requirements from reusable capabilities
  - Increase understanding of software frameworks
  - Raise awareness of the importance of interface designs
- Recognize and award creative contributors



# Open Source Software

## Bottom-up and Top-down

*Remember Alan Turing and John von Neumann!*

- Top-down
  - The application perspective: outside-in
  - Course grained reusability
- Bottom-up
  - The capability perspective: inside-out
  - Fine grained reusability
- The interface criteria

Reuse occurs when the user of the interface changes.
- Engineering model
  - The crate
  - The components

